The FUEL code project

James C. Osborn

Argonne Leadership Computing Facility

Lattice 2014

June 23-28

Columbia University, NY

FUEL project

- Initial goal:
  Use high level scripting language to wrap existing LFT libraries with focus on providing routines necessary for lattice generation

- Want to be as flexible/general as possible
  (arbitrary Nc, dimensions, etc.)

- Initial implementation over QOPQDP library (+ QDP/C, QLA, QIO, QMP), developed under US SciDAC program

- Steadily improving support for more actions and more analysis capability with goal of being a general LFT application

Scripting language

- High level, easy to create & manipulate data structures, garbage collection
- Provides ability to easily modify program without recompiling
- Can quickly implement and test new routines
- No need for complex input file formats (script is input file)
- Can include scheduler job scripts/workflow logic in input scripts
- Scripting language requirements
  - Small
  - Easy to use
  - Easy to port
  - Easy to embed and interface with existing or new libraries

Lua

- Small, simple, fast and powerful scripting language
- Developed at Computer Graphics Technology Group (Tecgraf) at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), in Brazil
- Name means “moon” in Portuguese
- About 17k lines of ANSI C (easily ported)
- Designed to be embedded and easily interface with C libraries
- Liberal MIT license

FUEL project

- FUEL: Framework for Unified Evolution of Lattices (name used for marketing purposes)
- Code is currently called qhmc
- Code development: http://github.com/jcosborn/qhmc
- Tarballs: http://usqcd-software.github.io/downloads/qhmc
- No official website/documentation yet (should have before proceedings are out)

FUEL project

- Being used by members of Lattice Strong Dynamics collaboration
- Other contributors:
  - Meifeng Lin (BNL): stout smearing, MG-HMC
  - Evan Weinberg (BU): gauge fixing, staggered spectrum
- Requires QMP, QIO, QLA, QDP, QOPQDP
  (many additional contributors)
- Easiest way to install is use “qinstall” script
  http://github.com/usqcd-software/qinstall

US Lattice QCD SciDAC libraries

level 3:

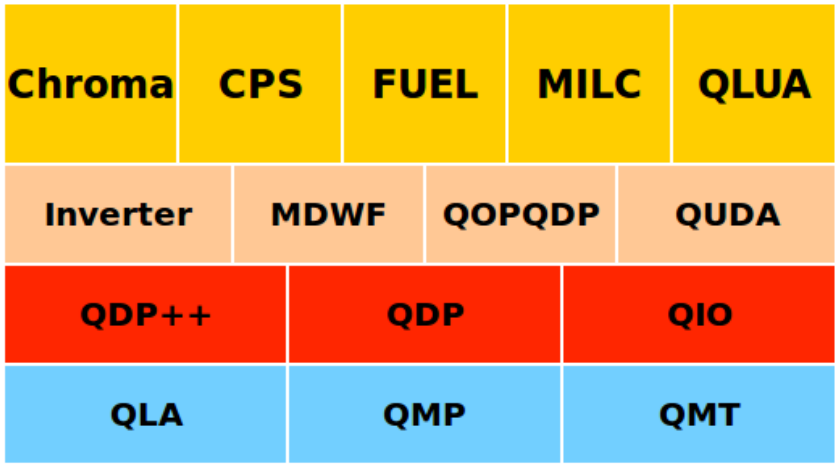
- Optimized versions of common routines

level 2:

- QIO I/O
- QDP data parallel

level 1:

- QMP message passing
- QLA linear algebra



FUEL (qhmc) current status

- Most routines come from QOPQDP:

  Full support for Nc=1,2,3,N versions of most routines
  - Gauge action, force, heatbath
  - Asqtad smearing, solver, force
  - HISQ smearing, solver, force (Nc=3 in QOPQDP, Nc≠3 in FUEL)
  - Wilson clover solver, multigrid
  - Plain Wilson force
  - (Plain) Domain Wall solver

  Missing
  - Clover force
  - DW variants and force
  - Eigenvalue related routines (Lanczos, Rayleigh-Ritz) have codes in QDP, need to be moved

FUEL (qhmc) current status

- Staggered support fairly complete
  - Actions: asqtad, HISQ, stout, nHYP, ...
  - Meson, baryon spectrum
- Wilson support improving
  - Plain wilson HMC, clover solver only
  - Multigrid solver
  - Meson, baryon spectrum
- Coulomb gauge fixing, Wilson loops, Wilson flow, ...
- HMC integrator, parameters controlled from Lua
  - Very flexible, easy to tune (at runtime)

FUEL (qhmc) current status

- Extensive, flexible gauge smearing
- Several routines available
  - staples, fat7, exponential, product, projections (unitary, traceless anti-Hermitian), etc.
- All with derivatives, chain rule
- Standard smearings (stout, nHYP) are built from these (asqtad and HISQ also have built in support)
- Can build arbitrary new combinations in run scripts

FUEL (qhmc) current status

- Wilson spectrum code under development
- No native "Dirac Propagator" (4Nc x 4Nc matrix field)
- Has "Dirac Fermion" (4Nc vector)
  - can be used for simple analysis
  - has example creating Dirac prop. out of 4Nc Dirac fermions, with gamma matrix multiplies across 4Nc vectors
- New version creates Dirac prop. as 4 x 4 matrix of “Color Matrix” field
  - does "virtual" multiply by gamma matrix (stores coefficients)
  - has routines for meson & baryon 2pt functions
  - should be capable of (most?) other analysis

Caveat emptor

- Currently used in production
- Still fairly new and rapidly evolving
  - Interfaces may change
  - No major backward compatibility breaks planned
- Some regression tests, but not extensive
- Not much documentation
  - Basically just existing scripts
- Existing scripts mainly designed for testing code
  - Not very user friendly
  - New user scripts in development

FUEL new Lua scripts

```
L = Lattice{4,4,4,8}
G = L:GaugeField{group="SU",nc=3}
G:Load("lattice_file_name")
GA = GaugeAction{kind="plaquette",beta=6,field=G}
M = G:Momentum()
I = Integrator{kind="leapfrog",action=GA,field=G,
               momentum=M,tau=1,nSteps=40}
E = Evolver{kind="HMC",integrator=I}
E:Run()
```

FUEL new scripts status

- Gauge action/integration done
  - Need to add existing gauge smearing and fermion actions
- Supports dynamic (runtime) Nc and compile-time
  - Can create multiple gauge fields of different groups
    (still needs routines for combining them)
  - Runtime Nc currently through 'N' color QOPQDP/QDP/QLA
    (specific Nc versions called for runtime Nc=1,2,3; not complete)

Related work

- Qlua:
  - Started by Andrew Pochinsky (MIT)
  - Fairly complete Lua wrappers for QDP/C
  - Mainly used for analysis
  - Adding some support for HMC and multigrid
  - Has support for some libraries (MDWF) not yet in FUEL
- Originally mostly orthogonal design & goals to FUEL
  - Starting to see more overlap
  - Investigating possible ways to allow codes to coexist

FUEL current plans

- Finish new scripts
  - Gauge smearing
  - Existing fermion actions
- Add new actions
  - Clover HMC, DW
  - BSM related actions (naïve implementation of higher representations)
- Continue integration/development improved solvers (MG)
- Integrate other level 3's
  - QUDA, MDWF, BG/Q code, Phi code